

DYNAMIC INFILLING ANCHORS FOR FORMAT-CONSTRAINED GENERATION IN DIFFUSION LLMs

Anonymous authors

Paper under double-blind review

ABSTRACT

Diffusion large language models (dLLMs) have recently emerged as a compelling alternative to autoregressive LLMs, offering bidirectional attention and parallel sequence generation. These properties allow dLLMs to exploit global contextual information and naturally support the integration of non-sequential constraints, making them particularly suitable for format-constrained tasks such as generating parseable JSON or reasoning-answer templates. A straightforward approach is to enforce such constraints with fixed anchors, but this often results in rigid generation spans, leading to truncated reasoning or redundant content. To overcome this limitation, we propose a training-free method, Dynamic Infilling Anchors (DIA). DIA dynamically adjusts generation length by estimating appropriate end-anchor positions before content generation, followed by iterative infilling between anchors. This flexible mechanism ensures structural correctness and semantic coherence while avoiding the inefficiencies of fixed-span methods. Experiments on reasoning-oriented benchmarks demonstrate that DIA substantially improves both format compliance and answer accuracy, achieving significant gains on GSM8K and MATH under zero-shot settings. These results highlight the promise of dLLMs for reliable, structure-aware generation and establish DIA as a practical pathway toward robust format-constrained text generation.

1 INTRODUCTION

In recent years, diffusion large language models (dLLMs) (Nie et al., 2025; Ye et al., 2025; Labs et al., 2025; Song et al., 2025; Deepmind, 2024) have attracted increasing attention due to their distinctive computational mechanisms and promising potential. Unlike traditional autoregressive language models (AR LLMs), which rely on left-to-right sequential decoding, dLLMs are not restricted to unidirectional dependencies during generation. Instead, they employ a bidirectional attention mechanism, enabling the model to update token representations at each step by leveraging complete contextual information simultaneously. This mechanism allows all positions in a sequence to be predicted in parallel rather than generated step by step, thereby substantially enhancing both modeling flexibility and computational efficiency. Beyond efficiency gains, this parallelism also strengthens the contextual modeling capacity of dLLMs, enabling them to capture global dependencies more comprehensively.

Within this property, we identify not only the potential to enhance contextual modeling and generation efficiency, but also the possibility of directly incorporating non-sequential constraints into the generation process. The exposure of a fully masked sequence in dLLMs allows us to impose global constraints on the target output by directly editing the masked sequence. For instance, one may preemptively replace selected mask tokens with predetermined conclusions or mandatory content, thereby guiding the model toward iterative optimization under the specified requirements. This observation motivates us to explore the application of dLLMs to the problem of format-constrained generation. The term refers to scenarios in which the model’s output must strictly adhere to pre-defined structures and requirements. For example, producing parseable JSON representations. To evaluate this capability, we adopt a representative thinking-answering task as the testing scenario, where existing dLLMs fail to achieve satisfactory outcomes.

To address these challenges, a straightforward approach is to enforce structural constraints by inserting anchors (e.g. `< think >`, `< /think >`, `< answer >`, `< /answer >`) directly into the masked

sequence. However, while this approach appears intuitive, it also introduces new challenges. Once anchor positions are fixed in advance, the generative space between them becomes rigid, forcing the model to allocate tokens within predetermined boundaries. Such rigidity can lead to suboptimal allocation of generative space and ultimately impair output quality. In practice, when the fixed span between anchors is too short, the reasoning process is often truncated before completion. On the other hand, when the span is too long, the model tends to produce redundant or repetitive content, thereby reducing both efficiency and reliability.

To obtain an appropriate generation length between anchors, thereby ensuring format correctness while maintaining generation quality, we propose a more flexible training-free strategy termed *Dynamic Infilling Anchors (DIA)*. Our approach is inspired by previous studies on dLLMs(Li et al., 2025), which demonstrates that the model can estimate the position of the end token with only one or a few prediction steps, thereby determining a suitable generation length. We extend this capability to predict the proper positions of anchors before content generation. Specifically, our method consists of two stages: (1) generation length adjustment by estimating position of the end anchor, and (2) iterative generation between fixed anchors.

The first stage of our method involves adjusting the generation space by estimating the position of the end anchor. Following the user prompt, the model initializes a relatively short, fully masked sequence, which serves as a starting point for the task output length and is dynamically extended later. For a think-answer task, this masked sequence is evenly divided into two blocks, with the corresponding begin anchors inserted at the start of each block. We then determine the anchor positions sequentially, one block at a time. Within each block, the model performs a single prediction step on the sequence, which is prefilled with the begin anchor. If the prediction fails to produce an end anchor or yields one with insufficient confidence, it suggests that the current generation length is inadequate. Therefore, we extend the block by appending additional masked tokens to ensure adequate space for content generation and repeat the prediction step. This extension continues until the model successfully produces a valid end anchor or the block length reaches its upper limit. The design of Stage I fully leverages the model’s awareness of the generation space; it guarantees sufficient allocation for each phase while minimizing redundant space and unnecessary computation.

The second stage performs iterative generation after anchors are fixed. In the previous stage, we obtained a reasonable generation length and fixed the position of the end anchor. Based on this setup, we now generate the intermediate content between the anchors. This step effectively compensates for the limitations of single-step prediction and helps the model establish clear semantic boundaries across different segments, thereby promoting coherent content generation.

We validate the effectiveness of DIA on reasoning-oriented benchmarks. Experimental results on GSM8K(Cobbe et al., 2021) (0-shot) and MATH(Hendrycks et al., 2021) (0-shot) show that our method improves format correctness from 58.83% and 29.10% to **72.63%** and **76.82%**, respectively. Moreover, by better controlling the generation space, our method also improves answer accuracy from 14.86% and 19.52% to **46.78%** and **20.08%**, respectively. These results demonstrate that DIA substantially enhances both the reliability and quality of format-constrained generation with dLLMs. In summary, our contributions are three-fold:

1. We introduce a novel dLLM-based strategy for format-constrained generation.
2. We design a dynamic adjustment mechanism that flexibly allocates generative space, mitigating the rigidity of fixed-anchor methods.
3. We will release code and resources to foster reproducibility and further research in this emerging area.

2 RELATED WORKS

Diffusion Large Language Models The evolution of diffusion paradigms in language modeling can be traced back to masked language models(Devlin et al., 2019), which randomly mask a subset of tokens in the input and predict the missing content, laying the groundwork for denoising-based generation. Building on this idea, early studies introduced continuous-space diffusion language models(Jo & Hwang, 2025), mapping text into continuous latent representations and generating sequences through diffusion and reverse denoising. However, such methods suffered from ambiguity in representation and instability in decoding discrete text. To address this limitation, discrete-space

diffusion language models(Austin et al., 2023) were proposed, directly modeling diffusion and denoising at the token level, thereby aligning the process more naturally with the discrete nature of language. Along this trajectory, BlockDiffusion(Arriola et al., 2025) incorporated block-wise modeling to mitigate the computational inefficiencies of diffusion-based text generation. For large-scale pre-training, a practical strategy for diffusion large language models (dLLMs) is to initialize them from pretrained autoregressive models(Gong et al., 2025a; Ye et al., 2025) and further align them with instructions to enhance task adaptability(Yang et al., 2025b; You et al., 2025; Song et al., 2025). To strengthen advanced capabilities, researchers have also explored reinforcement learning(Wang et al., 2025; Zhao et al., 2025; Gong et al., 2025b) as a post-training method. Meanwhile, dLLMs are being extended to multimodal scenarios through cross-modal alignment, enabling broader applications in understanding and generating modalities such as images and speech.

Format-Constraints Format-constrained generation is critical for deploying language models, as it directly affects the parseability and reliability of code generation, structured outputs, and reasoning templates. Existing studies often constrain the input side (prompt design(Ye et al., 2024) and example-based guidance(Min et al., 2022)), yet they are unstable under long-chain or high-complexity reasoning; output-side repair (post-processing and re-ranking(Gao et al., 2025; Zhuang et al., 2025)) improves format compliance but struggles to preserve semantic and structural consistency simultaneously. Fine-tuning or reinforcement learning on task-specific data(Song et al., 2025; Xiong et al., 2023; Cui et al., 2024; Yang et al., 2023) can enhance robustness, but the approach is costly and generalizes poorly across tasks. Constrained decoding(Mündler et al., 2025; Banerjee et al., 2025) with grammars or finite-state machines enforces strict compliance at the expense of efficiency and flexibility.

Large Language Models The evolution of LLMs(Yang et al., 2025a; Grattafiori et al., 2024; DeepSeek-AI et al., 2025; Anthropic, 2025; Deepmind, 2025; xAI, 2025; OpenAI, 2025) has been fundamentally driven by insights from scaling laws (Kaplan et al., 2020), which reveal power-law relationships among model size, data, and compute, thereby guiding systematic capability improvements. Building on this foundation, researchers have observed the emergent phenomenon of in-context learning (ICL)(Min et al., 2022), whereby LLMs can rapidly adapt to new tasks from demonstrations without explicit parameter updates, showcasing remarkable transfer and generalization abilities. To further enhance practical usability and alignment with human preferences, post-training techniques such as fine-tuning(Ouyang et al., 2022) and reinforcement learning(Schulman et al., 2017; Rafailov et al., 2024; Shao et al., 2024) have been extensively applied, playing a central role in task adaptation and alignment. In parallel, the rise of multimodal models has spurred advances in cross-modal alignment(Li et al., 2023; Liu et al., 2023), enabling LLMs to operate effectively across text, vision, and speech, and thereby extending their versatility. Collectively, these lines of research have driven sustained progress in LLM capability, alignment, and applicability.

3 METHOD

3.1 PRELIMINARY

Inference of dLLMs. In the generation stage of a diffusion language model (dLLM), the response sequence to be refined is initialized by concatenating the input prompt with a fully masked sequence of a specified length:

$$x_t = \text{Concat}(\text{prompt}, \{[MASK]\}_{0: \text{max.len}-1}), \quad (1)$$

where max.len denotes the fixed response length. The generation process follows a discrete-time masked diffusion procedure, which can be formulated as a Markov chain. Thus, each prediction step depends only on the previous state, and in every iteration only the masked positions are updated in parallel:

$$P_{0|t} = \prod_{s=t}^0 \prod_{i=0}^{n-1} P_{s|s+1}(x_s^i | x_{s+1}), \quad (2)$$

$$P_{s|s+1}(x_s^i | x_{s+1}) = \begin{cases} 1 & \text{if } x_{s+1}^i \neq [M], \text{ then } x_s^i = x_{s+1}^i, \\ 1 - \max(q(x_s^i)) & \text{if } (x_{s+1}^i = [M], \max(q(x_s^i)) < C), \\ \max(q(x_s^i)) & \text{then } x_s^i = [M], \\ & \text{if } (x_{s+1}^i = [M], \max(q(x_s^i)) \geq C) \\ & \text{or } s = 0, \text{ then } x_s^i \neq [M]. \end{cases} \quad (3)$$

Where $[M]$ denotes the $[MASK]$ token, $q(x_s^i)$ represents the output logits at position i in step s , and C is the minimum confidence threshold.

3.2 DYNAMIC INFILLING ANCHOR

To overcome the limited flexibility of straightforward infilling methods in diffusion language models, we propose DIA, a training-free, two-stage approach. DIA selects an appropriate end-anchor position through a single-step prediction, thereby ensuring both format correctness and generation quality. The overview of our method is illustrated in Figure 1.

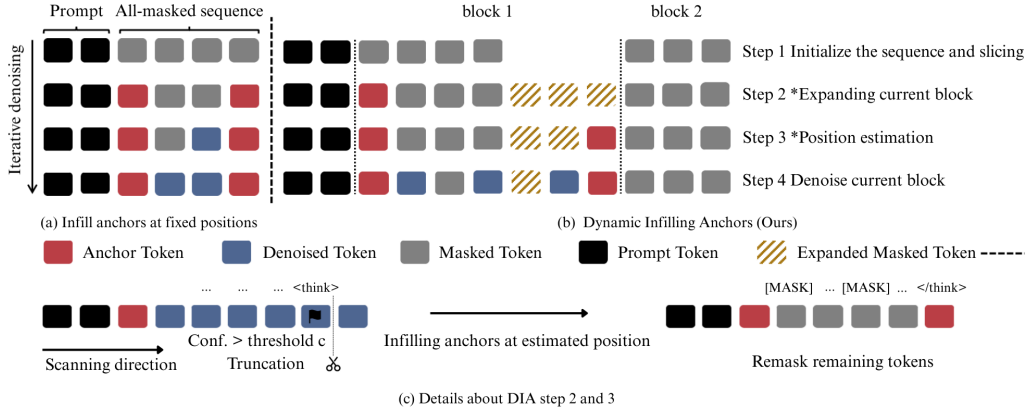


Figure 1: Dynamic Infilling Anchors (DIA). (a) Fixed-position infilling baseline. (b) Overview about our methods: DIA. (c) Details of expansion and anchor infilling steps with truncation and remasking.

3.2.1 GENERATION LENGTH ADJUSTMENT BY ESTIMATING POSITION OF THE END ANCHOR

DLLMs implicitly acquire a prior distribution over response termination positions from large-scale training corpora (Li et al., 2025). Specifically, for different input queries, the confidence of predicting the eos token at various positions within the answer sequence is not uniform, but instead exhibits a trend correlated with the appropriate response length. Building on this insight, we extend this capability to format-constrained tasks. For a typical reasoning-answer task, when the model receives the start anchor of a reasoning or answering section, it should be able to anticipate at what sequence length a corresponding “end-of-reasoning” or “end-of-answering” anchor is likely to occur. Intuitively, if the allocated generation space is sufficient to accommodate the reasoning or answering process, the one-step prediction will contain an end anchor (or partial end anchor) with high confidence exceeding a given threshold. Conversely, if the generation space is insufficient, the corresponding anchor will either fail to appear or appear only with substantially reduced confidence.

Building on this assumption, we design the generation-space estimation procedure of DIA. Given an input sequence X , which consists of the user query Q and a fully masked sequence X_L of a specified length L , DIA divides the sequence into two blocks ($\mathcal{C} = \{C_1, C_2\}$) of equal size (in terms of masked tokens), corresponding to the reasoning and answering stages. For each block, DIA first pre-fills the start anchor at the beginning of the decodable region. After inserting the start anchor, the block undergoes a one-step prediction. The prediction results and their associated confidence scores are used to determine whether the allocated generation length is appropriate. Since the model is

unlikely to produce a complete anchor token sequence in a single prediction, partial anchors are also incorporated into the decision mechanism. If the prediction either fails to produce an end anchor (or a partial end anchor) or yields an end anchor with confidence below the threshold c , the length of current block is expanded by a fixed length Δ , and the 'predict-decide' cycle is repeated until the generation space is sufficient to support the model in completing the reasoning or answering process. When multiple positions in the sequence satisfy the confidence threshold simultaneously, we retain the position closest to the left boundary to prevent the generation of duplicate end anchors within the sequence. To avert unbounded expansion, a maximum block length M is imposed. We truncate the redundant tokens following the selected end-anchor position and subsequently complete the partial end anchor to form a full one.

3.2.2 ITERATIVE DENOISING WITH INFILLING

In Stage I, we establish the block boundaries by determining the positions of the anchors. Based on these fixed semantic boundaries, the model then iteratively generates the intermediate content within the block. The fixed anchors serve as guidance, ensuring clear separation between segments and thereby promoting coherent content generation.

We process the blocks sequentially through two stages. Specifically, once the length of the thinking block is determined, its content is generated; the additional information obtained from this reasoning step is then used to determine the length of the answering block, which is subsequently generated in an iterative manner. This design maximizes the benefit of the reasoning process by leveraging the information gained in the first stage to enhance the quality of the final answer. Further implementation details are provided in Algorithm 1.

Algorithm 1 DIA

Require: Input sequence $X = \{Q, X_L\}$, begin-anchor set $\mathcal{B} = \{b_1, \dots, b_{|\mathcal{B}|}\}$, end-anchor set $\mathcal{E} = \{e_1, \dots, e_{|\mathcal{E}|}\}$, confidence threshold c , expand size Δ , max length M

Ensure: Completed sequence $X = \{Q, C_1, \dots, C_{|\mathcal{B}|}\}$

- 1: Divide X_L into $|\mathcal{B}|$ blocks $\mathcal{C} = \{C_1, \dots, C_{|\mathcal{B}|}\}$, each of max length M
- 2: **for** $i \leftarrow 1$ to $|\mathcal{B}|$ **do**
- 3: Insert begin anchor b_i at the head of block C_i
- 4: **end for**

Stage 1: Generation length adjustment by estimating position of the end anchor

- 5: **for** each block C_i **do**
- 6: **while** True **do**
- 7: $Y \leftarrow \text{Infer}(Q, C_1 \dots C_i)$ *perform one diffusion-based inference*
- 8: Scan tokens of C_i from head
- 9: **if** a subsequence $y \subseteq Y$ matches some part of $e_i \in \mathcal{E}$ with $\text{Conf}(y) > c$ **then**
- 10: Truncate C_i at this position
- 11: **break**
- 12: **else if** no partial match found and $|C_i| + \Delta \leq M$ **then**
- 13: Expand C_i by Δ tokens *controlled by expand size*
- 14: **else if** $|C_i| + \Delta > M$ **then**
- 15: Stop expansion for C_i
- 16: **break**
- 17: **end if**
- 18: **end while**
- 19: Infill selected $e_i \in \mathcal{E}$ at the tail of C_i

Stage 2: Iterative Denoising with Infilling

- 20: Generate all remaining masked positions in C_i using $\text{Infer}(Q, C_1 \dots C_i)$
- 21: **end for**
- 22: **return** $X = \{Q, C_1, \dots, C_{|\mathcal{B}|}\}$

4 EXPERIMENTS

4.1 BENCHMARKS

To systematically evaluate the effectiveness of our method, we adopt two reasoning-sensitive mathematical benchmarks: GSM8K 0-shot and MATH 0-shot. **GSM8K**(Cobbe et al., 2021) is a widely used dataset of grade-school math word problems, covering basic arithmetic and commonsense reasoning tasks, and thus serves as a reliable measure of a model’s performance in everyday numerical reasoning scenarios. In contrast, **MATH**(Hendrycks et al., 2021) is a more challenging benchmark that spans competition-level problems from elementary to advanced mathematics, encompassing diverse problem types and difficulty levels, thereby providing a rigorous assessment of a model’s capabilities in complex reasoning and knowledge generalization.

4.2 BASELINES

We select Dream-7B-Base-v0 and Dream-7B-Instruct-v0 as our baseline models. The Dream-7B series is initialized from the Qwen model family and has achieved superior performance compared to other open-source diffusion models on multiple benchmark tasks. To ensure fairness, all experiments are conducted with corresponding modifications to the official codebase, without applying any additional acceleration or optimization techniques.

4.3 IMPLEMENTATION DETAILS

Our method is implemented within the PyTorch framework. For a fair comparison, all models are evaluated under the same GPU configuration when tested on identical tasks. Additional implementation details are provided in Appendix B.

4.4 MAIN RESULTS

We conduct a comprehensive evaluation on the two benchmarks. Table 1 reports the comparison between our method and the baselines. Specifically, Dream-7B-Base-v0 and Dream-7B-Instruct-v0 generate responses by relying solely on additional format-constrained prompts. In contrast, the infilling approach inserts the corresponding anchors at designated positions within the response sequence of Dream-7B-Base-v0, thereby guiding the model to produce answers.

We introduce two metrics, Format Score S_{format} and Accuracy $Acc.$, for evaluation. Accuracy measures whether the generated response is correct, while Format Score assesses whether the response adheres to the predefined format requirements.

Table 1: Comparison of Methods on Format Adherence and Benchmark Performance. DIA achieves the highest format scores across both GSM8K and MATH, substantially outperforming baseline and infilling approaches. These results highlight the robustness and effectiveness of DIA in enforcing strict structural constraints while maintaining competitive answer accuracy.

	0-shot GSM8K		0-shot MATH-500	
	S_{format}	$Acc.$	S_{format}	$Acc.$
Dream-7B-Base (Ye et al., 2025)	0	68.99	0	25.14
Dream-7B-Instruct (Ye et al., 2025)	0	15.01	0	25.28
Infilling Baseline	58.83	14.86	29.10	19.52
Dynamic Infilling Anchor (Ours)	72.63	46.78	76.82	20.08

Compared to the performance degradation introduced by the infilling approach, DIA achieves superior results in both format adherence and answer quality. On GSM8K, DIA not only raises the format score from 58.83% to 72.63% but also substantially improves accuracy from 14.86% to 46.78%, highlighting its ability to simultaneously enforce structural fidelity and enhance reasoning correctness. On the more challenging MATH benchmark, DIA boosts the format score from 29.10% to 76.82%, demonstrating remarkable robustness in preserving structural anchors even under complex

problem settings, while maintaining comparable answer accuracy to baseline methods. The results clearly demonstrate that DIA addresses the shortcomings of baseline models and methods under format-constrained tasks, ensuring accurate preservation of the required format. Moreover, unlike the infilling approach, DIA’s flexible design of generation length allows each stage to maintain high answer quality, thereby achieving a better balance between performance and format correctness across diverse benchmarks.

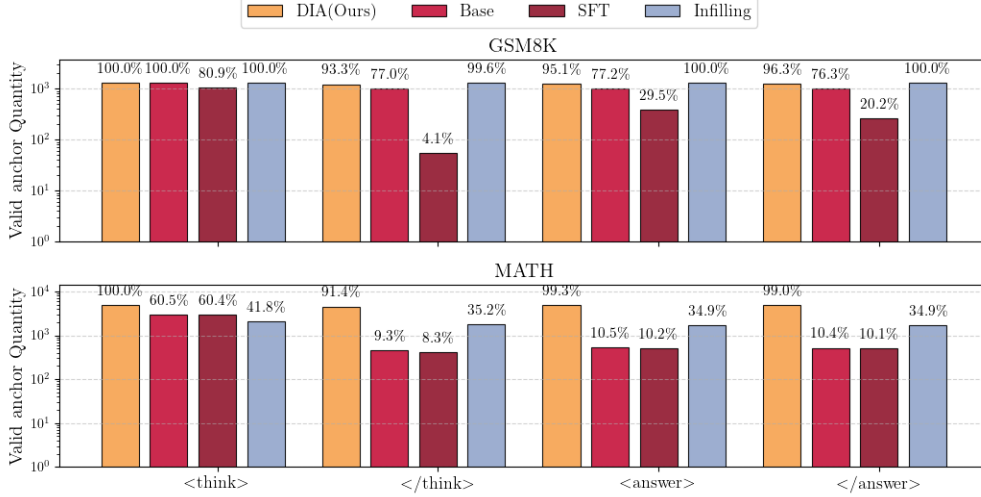


Figure 2: DIA delivers reliable anchor preservation and stable performance across different benchmarks. Even as task complexity increases on the more challenging MATH, DIA consistently maintains high anchor retention, underscoring its robustness under stricter reasoning and formatting requirements.

Figure 2 presents a detailed comparison of anchor retention ratios across different methods. Overall, DIA demonstrates outstanding stability on both GSM8K and MATH, consistently achieving nearly 100% retention across all four anchors, including both begin anchor (< think > and < answer >) and end anchor (< /think > and < /answer >). This robust performance shows that the proposed two-stage generation strategy not only preserves anchors under varying conditions but also enforces strict compliance with the predefined format throughout the entire sequence. Such stability is particularly important in reasoning-oriented tasks, where structural deviations can lead to incomplete, unparseable, or misleading outputs.

In contrast, the Base and SFT models suffer from significant structural degradation. For example, on GSM8K, their retention rates for < /think > collapse to only 4.4% and 29.5%, respectively, and on MATH, the rates for < /think > and < /answer > drop to single digits. These results reveal a consistent failure of conventional methods to maintain boundary integrity, especially in longer or more complex reasoning chains, where models tend to lose track of global structure and generate unbalanced outputs. Such issues undermine the reliability of the generated content and illustrate why relying solely on prompt-based constraints or fine-tuning strategies is insufficient for strict format adherence.

Although the Infilling baseline achieves higher anchor retention than Base and SFT—nearly matching DIA on GSM8K for < think > and < answer >, its performance on begin anchors remains unstable. Crucially, this preservation does not translate into gains in overall format correctness or answer accuracy. For instance, while Infilling retains anchors on GSM8K, its downstream results remain far below DIA in both structural and semantic evaluations. This mismatch highlights that simply inserting anchors is not enough; without a dynamic mechanism for allocating and regulating generation space, models either over-generate redundant tokens or fail to stop at the correct boundaries.

Taken together, these results provide a fine-grained validation of Table 1. They show that DIA not only outperforms existing approaches in aggregate metrics but also secures overwhelming superiority in preserving critical anchors across diverse datasets. By ensuring that every anchor is faithfully

retained, DIA substantially enhances the reliability of format-constrained generation, laying a foundation for robust application of dLLMs in reasoning, structured reporting, and other scenarios where strict adherence to format is essential.

4.5 ANALYSIS EXPERIMENTS

4.5.1 BEHAVIOR OUTSIDE ANCHOR CONTEXTS

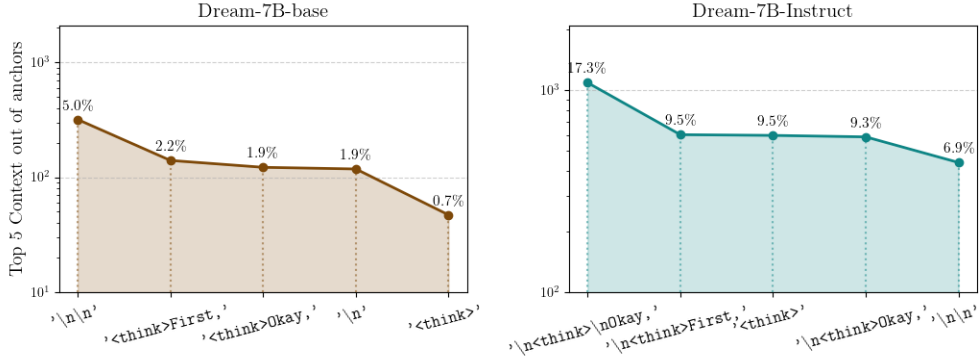


Figure 3: Top-5 statistics of out-of-anchor content generated by baseline models across different benchmarks. The baseline models fail to establish effective semantic boundaries aligned with anchor positions, leading to unconstrained content generation.

To more comprehensively evaluate model performance under format-constrained tasks, we conducted a statistical analysis of the responses generated by Dream-7B-Base-v0 and Dream-7B-Instruct-v0 on two benchmarks. Our analysis focuses on the content appearing beyond the `</answer>` anchor boundary, as this indicates whether the models can effectively leverage the semantic boundaries established by anchors to properly constrain their generation. Specifically, we examined all responses across the two benchmarks and extracted the Top-5 most frequent continuations occurring after the `</answer>` boundary for both models, with the results shown in Figure 3. This analysis provides a fine-grained perspective on boundary robustness, complementing aggregate metrics by revealing how models behave when the intended termination point has already been reached.

As shown in Figure 3, Dream-7B-Base-v0 produces dispersed and low-frequency redundancy beyond the `</answer>` anchor, with all Top-5 patterns below 6%, whereas Dream-7B-Instruct-v0 exhibits more concentrated redundancy, with Top-5 patterns reaching up to 17.3% and dominated by repeated `< think >` tokens. The contrast highlights that the Base model tends toward uncontrolled drifting, while the Instruct variant systematically re-enters the reasoning phase, reflecting a structural weakness in anchor boundary enforcement. Overall, the Base model lacks effective boundary control, while the Instruct model suffers from patterned continuations, and both fail to reliably terminate at the anchor—underscoring the necessity of DIA in eliminating out-of-anchor redundancy and ensuring format adherence. Importantly, such failures not only compromise readability but also propagate errors to downstream applications that rely on strictly bounded outputs.

4.5.2 EXPAND TIMES

To establish a reasonable upper bound for the maximum block length, we analyzed the number of extensions in the reasoning part of all format-correct responses. The details are presented in Figure 4. The results show that the chosen maximum length threshold effectively ensures the allocation of appropriate generation space. Specifically, the observed extension counts fall within the range of (30, 85), which is substantially smaller than the number of extensions permitted by the maximum threshold. In other words, although a large upper bound is allowed, the vast majority of responses naturally converge to a much smaller range of expansions, confirming that the setting of the maximum block length is both sufficient and not overly restrictive.

Moreover, the proportion of format-correct responses within this range consistently exceeds 90%, further validating both the effectiveness of our method and the appropriateness of the current threshold setting. Importantly, this trend is observed across both GSM8K and MATH, where over 94% of samples fall into the effective expansion range, indicating that DIA adapts reliably to tasks of different scales and difficulties. Such stability suggests that the block-length constraint not only prevents degenerate over-expansion but also preserves high-quality structural adherence across benchmarks.

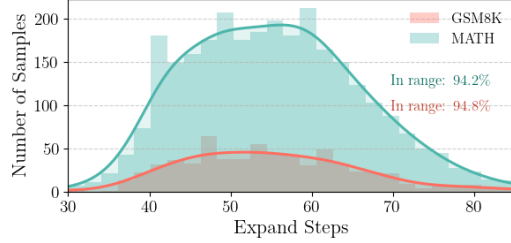


Figure 4: Statistics of effectively expanded samples. The maximum length threshold ensures that the vast majority of cases receive an appropriate number of expansions, thereby safeguarding answer quality.

5 DISCUSSIONS

While Dynamic Infilling Anchors (DIA) have shown strong effectiveness in improving format-constrained generation, several limitations remain. First, the current method relies on manually specified anchors, assuming that task boundaries and semantic roles (*e.g.*, reasoning vs. answering) are fixed. In more complex tasks such as open-domain dialogue or multi-stage reasoning, anchors may not follow stable positions, and their semantics may shift with context. Extending DIA to automatically infer or adapt anchor definitions remains an open challenge. Second, DIA introduces inference overhead from iterative space adjustments. Although modest in our experiments, scaling to longer outputs or interactive systems may require more efficient mechanisms for anchor prediction and length control. Finally, our evaluation is limited to reasoning datasets like GSM8K and MATH; whether DIA generalizes to long-form writing, program synthesis, or multimodal tasks is still uncertain.

Despite these limitations, the anchor-based perspective also suggests new opportunities. Anchors need not be restricted to reasoning and answering boundaries; they could represent higher-level structures such as proof steps, code blocks, or multimodal transitions. In tasks where anchor semantics evolve dynamically, DIA could be extended with adaptive or hierarchical anchor systems that refine themselves during generation. Combining DIA with lightweight training or alignment methods may also help models acquire richer anchor semantics, reducing manual specification while improving robustness across domains. These possibilities suggest that anchors are not only a control mechanism for current benchmarks but also a lens for rethinking how diffusion language models manage structure and meaning in more complex generative scenarios.

6 CONCLUSION

In this work, we introduced Dynamic Infilling Anchors (DIA), a training-free method for enhancing format-constrained generation in diffusion language models (dLLMs). By adopting a two-stage decoding strategy—length expansion guided by end-anchor prediction, followed by explicit anchor completion and content generation—DIA achieves a strong balance between structural fidelity and semantic quality. Experiments on reasoning benchmarks such as GSM8K and MATH show that DIA substantially improves format adherence while maintaining competitive answer accuracy, validating its effectiveness in tasks requiring both precision and reliability.

Beyond these empirical results, our study underscores the broader potential of dLLMs for structured text generation without additional training. Leveraging their intrinsic awareness of generation space, DIA demonstrates that such models can be steered to meet strict output constraints, opening opportunities in code generation, structured proofs, and machine-readable reporting. At the same time, DIA points to several avenues for future research, including automated anchor design, extensions to hierarchical or nested constraints, and integration with lightweight training or alignment techniques. By bridging structural control and semantic quality, DIA lays the groundwork for deploying dLLMs as dependable reasoning assistants and as general-purpose models in real-world applications where consistency, interpretability, and reliability are paramount.

ETHICS STATEMENT

This research does not involve human participants, personally identifiable information, or sensitive user data. All experiments are carried out on publicly available datasets, namely GSM8K and MATH, which are widely adopted benchmarks for assessing the reasoning ability of large language models. These datasets consist of synthetic or anonymized problem–solution pairs and do not contain private or proprietary information, thereby avoiding risks associated with data collection or misuse.

As with any work involving large language models, we acknowledge potential ethical concerns. A risk of misuse remains: models augmented with improved format-adherence mechanisms could be applied in contexts where reasoning outputs are consumed without adequate verification, potentially leading to the dissemination of incorrect or misleading content. We emphasize that our proposed method should only be deployed in scenarios where outputs are subject to human oversight or rigorous automated validation. While our method focuses on structural control rather than altering model parameters, it inherits the biases and limitations of the underlying pretrained models. Although Dynamic Infilling Anchors (DIA) does not amplify such issues, users of this technique should remain mindful of the broader ethical challenges associated with large-scale language models.

We stress the importance of responsible deployment. The intention of this work is to advance research on reliable, format-constrained generation and to provide the community with a training-free mechanism that improves structural fidelity without additional fine-tuning or data collection. We do not endorse its use in high-stakes domains such as medical diagnosis, legal reasoning, or automated decision-making without extensive domain-specific evaluation and safeguards. By situating our contributions within established ethical guidelines in AI research, we aim to ensure that the benefits of this work are realized while minimizing risks of harm.

REPRODUCIBILITY STATEMENT

We have taken comprehensive steps to maximize the reproducibility of our results and to facilitate independent verification by other researchers. To this end, we adhere to several principles of transparent and replicable research practice.

The appendix provides extensive details about our experimental setup, including hardware configuration (GPU model, number of devices, and memory constraints), software environment (Python and PyTorch versions, dependencies, and library compatibility), parameter choices (generation hyperparameters, block sizes, anchor thresholds, and sampling strategies), and evaluation protocols (dataset splits, metrics, and scoring procedures). These details are presented to eliminate ambiguity and enable replication.

The implementation of Dynamic Infilling Anchors (DIA) is based on the official Dream-7B codebase with minimal modifications, all of which are clearly documented. We describe both algorithmic adjustments, such as the two-stage anchor-based procedure for length expansion and truncation, and essential engineering decisions including input preprocessing and memory optimization strategies. Such documentation lowers the barrier for others to replicate our findings without requiring extensive reverse engineering.

We also commit to releasing all necessary artifacts for reproduction. This includes the source code, scripts for data preprocessing and evaluation, and configuration files for running experiments. The release will be accompanied by instructions for environment setup and guidelines for reproducing results on GSM8K and MATH. Seed values and randomization settings will be provided to ensure consistent outputs across trials.

In line with community standards, long-term accessibility is emphasized. All released materials will be archived in a permanent public repository with version control, ensuring that they remain accessible as dependencies evolve. By combining detailed documentation, transparent reporting, and open resources, we aim to make our work fully reproducible and to encourage further extensions and critical evaluation by the research community.

REFERENCES

- Anthropic. Introducing claude 4, 2025. URL <https://www.anthropic.com/news/claude-4>.
- Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models, 2025. URL <https://arxiv.org/abs/2503.09573>.
- Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023. URL <https://arxiv.org/abs/2107.03006>.
- Debangshu Banerjee, Tarun Suresh, Shubham Ugare, Sasa Misailovic, and Gagandeep Singh. Crane: Reasoning with constrained llm generation, 2025. URL <https://arxiv.org/abs/2502.09061>.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model, 2024. URL <https://arxiv.org/abs/2306.16092>.
- Google Deepmind. Gemini diffusion: Our state-of-the-art, experimental text diffusion model, 2024. URL <https://deepmind.google/models/gemini-diffusion/>.
- Google Deepmind. Gemini 2.5: Our most intelligent ai model, 2025. URL <https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/>.
- DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Cheng-gang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojuan Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuan Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shut-ing Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanbiao Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL <https://arxiv.org/abs/2412.19437>.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL <https://arxiv.org/abs/1810.04805>.
- Jingtong Gao, Bo Chen, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. Llm4rerank: Llm-based auto-reranking framework for recommendations, 2025. URL <https://arxiv.org/abs/2406.12433>.
- Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models, 2025a. URL <https://arxiv.org/abs/2410.17891>.
- Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation, 2025b. URL <https://arxiv.org/abs/2506.20639>.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Paspuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rapparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collet, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenxin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein,

Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippas Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Rutu Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL <https://arxiv.org/abs/2103.03874>.

Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling, 2025. URL <https://arxiv.org/abs/2502.11564>.

- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL <https://arxiv.org/abs/2001.08361>.
- Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion, 2025. URL <https://arxiv.org/abs/2506.17298>.
- Jinsong Li, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jiaqi Wang, and Dahua Lin. Beyond fixed: Training-free variable-length denoising for diffusion large language models, 2025. URL <https://arxiv.org/abs/2508.00819>.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL <https://arxiv.org/abs/2301.12597>.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL <https://arxiv.org/abs/2304.08485>.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?, 2022. URL <https://arxiv.org/abs/2202.12837>.
- Niels Mündler, Jingxuan He, Hao Wang, Koushik Sen, Dawn Song, and Martin Vechev. Type-constrained code generation with language models. *Proceedings of the ACM on Programming Languages*, 9(PLDI):601–626, June 2025. ISSN 2475-1421. doi: 10.1145/3729274. URL <http://dx.doi.org/10.1145/3729274>.
- Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025. URL <https://arxiv.org/abs/2502.09992>.
- OpenAI. Introducing gpt-5, 2025. URL <https://openai.com/index/introducing-gpt-5/>.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL <https://arxiv.org/abs/2203.02155>.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL <https://arxiv.org/abs/2305.18290>.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, Yuwei Fu, Jing Su, Ge Zhang, Wenhao Huang, Mingxuan Wang, Lin Yan, Xiaoying Jia, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Yonghui Wu, and Hao Zhou. Seed diffusion: A large-scale diffusion language model with high-speed inference, 2025. URL <https://arxiv.org/abs/2508.02193>.
- Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models, 2025. URL <https://arxiv.org/abs/2509.06949>.

- xAI. Grok 4, 2025. URL <https://x.ai/news/grok-4>.
- Honglin Xiong, Sheng Wang, Yitao Zhu, Zihao Zhao, Yuxiao Liu, Linlin Huang, Qian Wang, and Dinggang Shen. Doctorglm: Fine-tuning your chinese doctor is not a herculean task, 2023. URL <https://arxiv.org/abs/2304.01097>.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL <https://arxiv.org/abs/2505.09388>.
- Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models, 2023. URL <https://arxiv.org/abs/2306.06031>.
- Ling Yang, Ye Tian, Bowen Li, Xincheng Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models, 2025b. URL <https://arxiv.org/abs/2505.15809>.
- Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models, 2025. URL <https://arxiv.org/abs/2508.15487>.
- Qinyuan Ye, Maxamed Axmed, Reid Pryzant, and Fereshte Khani. Prompt engineering a prompt engineer, 2024. URL <https://arxiv.org/abs/2311.05661>.
- Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning, 2025. URL <https://arxiv.org/abs/2505.16933>.
- Siyao Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning, 2025. URL <https://arxiv.org/abs/2504.12216>.
- Shengyao Zhuang, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. Rank-r1: Enhancing reasoning in llm-based document rerankers via reinforcement learning, 2025. URL <https://arxiv.org/abs/2503.06034>.

A NOTATION SUMMARY

Table 2: Notation Summary

Symbol	Description
$\bar{X} = \{Q, X_L\}$	Input sequence consisting of query Q and all-masked sequence X_L
Q	Input query provided to the diffusion-based LLM
X_L	Fully masked sequence to be partitioned into blocks
$\mathcal{B} = \{b_1, \dots, b_{ \mathcal{B} }\}$	Set of begin anchors inserted at block heads
$\mathcal{C} = \{C_1, \dots, C_{ \mathcal{B} }\}$	Set of blocks obtained from X_L
$\mathcal{E} = \{e_1, \dots, e_{ \mathcal{E} }\}$	Set of end anchors to be infilled at block tails
C_i	The i -th block after begin anchor insertion and expansion
$ C_i $	Current length of block C_i during expansion
y	Predicted subsequence scanned within block C_i
c	Confidence threshold for partial end-anchor detection
Δ	Expand size: number of tokens added per expansion step
M	Maximum allowed length of a block C_i
$\text{Infer}(X)$	Diffusion-based inference function applied to sequence X
$\text{Conf}(y)$	Confidence score assigned to subsequence y

B EXPERIMENTAL SETTINGS

Table 3: Experimental Settings

Category	Setting
Model	Dream-7B-Base / Dream-7B-Instruct
Datasets(Max new tokens)	GSM8K(256), MATH-500(512)
Evaluation Metrics	Format Score, Answer Score, Anchor Retention Rate
Confidence Threshold c	0.065, 0.05
Expand Size Δ	4, 4
Max Block Length M	512, 512
Diffusion Steps	512, 512
Batch Size	1, 3
Total Samples	GSM8K: 1319, MATH-500: 5000
Hardware	NVIDIA vGPU-32G, vGPU-48G
Framework	PyTorch 2.5.1, Python 3.10.16

C CASE STUDY

C.1 BASE MODEL

"query": "Brandon's iPhone is four times as old as Ben's iPhone. Ben's iPhone is two times older than Suzy's iPhone. If Suzy's iPhone is 1 year old, how old is Brandon's iPhone?", "method": "Base", "output": "<think> Suzy's iPhone is 1 year old. Ben's iPhone is two times older than Suzy's iPhone, so Ben's iPhone is 2 years old. Brandon's iPhone is four times as old as Ben's iPhone, so Brandon's iPhone is 4 years old. So the answer is 4 years old.</think><answer> 4 </answer>\n\n", "flaw": "Content out of anchor"

C.4 DIA

"query": "Charlie wants to sell beeswax candles. For every pound of beeswax, he can make 10 tapered candles. One pound of beeswax and the wicks cost \$10.00 in supplies. If he sells each candle for \$2.00 each, what is his net profit if he makes and sells 20 candles?", "method": "DIA", "output": ""<think>First, I need to figure out how much candles Charlie can make with 20 candles. Since he can make 10 candles from one pound of beeswax, he can make 20 candles from 2 pounds of beeswax. The pound of beeswax costs \$10, so 2 pounds will cost \$20. The total of supplies is \$20. Now, I need to calculate out how much money he will make from selling 20 candles. Since he sells each candle for \$2, he will make $20 \times \$2 = \40 . Now, I need to subtract the cost of supplies from the amount he will make from selling the candles. So, $\$40 - \$20 = \$20$. Therefore, Charlie will profit a net of \$20.00 and makes and sells 20 candles0</think><answer>20</answer>""", "flaw": "None."

"query": "John drives for 3 hours at a speed of 60 mph and then turns around because he realizes he forgot something very important at home. He tries to get home in 4 hours but spends the first 2 hours in standstill traffic. He spends the next half-hour driving at a speed of 30mph, before being able to drive the remaining time of the 4 hours going at 80 mph. How far is he from home at the end of those 4 hours?", "method": "DIA", "output": ""<think>First, I need to calculate the distance John traveled away. He drove for 3 hours at 60 mph, so the distance is $60 \times 3 = 180$ miles. Then, he turned around and tried to get home in 4 hours. He, spent the first 2 hours in standstill traffic, so he didn't cover any distance during that time. He, then spent the next half-hour driving at 30 mph, so the covered distance is $30 \times 0.5 = 15$ miles. He, spent the remaining 2 hours driving at 80 mph, so the covered distance is $80 \times 2 = 160$ miles. The, the total distance he covered while coming back is $15 + 160 = 175$ miles. Since, he traveled 180 miles away from home and then covered 175 miles back,,, he is distance from home at $180 - 175 = 5$ miles. the end of those 4 hours, he, he, 5555555555555555 miles5</think><answer>5 miles</answer>""", "flaw": "Generation length prediction not completely accurate."

D DLLM PROMPT TEMPLATE

Table 4: Example of Prompt Design

Field	Content
Instruction	You are a helpful assistant that helps the user to solve the question.
Output Format	You need to think first and then answer the question briefly by following the format:< think > ... < /think >< answer > ... < /answer >.
Input	Here are the questions: {QUESTION}

E USAGE OF LLMs

In accordance with the ICLR guidelines, we disclose the use of large language models (LLMs) in the preparation of this paper. LLMs were employed exclusively as a writing assistance tool for language polishing, grammar refinement, and improving readability. They were not involved in research ideation, experimental design, data analysis. All technical ideas, theoretical developments,

972 proofs, and experimental results presented in this paper are entirely the work of the authors. The
973 authors take full responsibility for the accuracy and integrity of the final submission.
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025